RSMS-4G Software Use Case Statements

There are four major software operations for the RSMS-4G: 1) providing logistical information, 2) performing measurements, 3) retrieving data, and 4) editing and testing script language routines.

1) **Providing Logistical Information :**

Logistical information provides logistical support for measurements, and data retrieval. The user provides information for initialization, equipment control connections, data paths, survey and measurement descriptions, coordinate location and bearing, and new component descriptions.

The user is given the option of *initializing* by starting a new data output file and/or removing all or some of the calibration data that is currently stored on disk.

At each new software startup, the user is given the option to confirm or change information about the *system configuration* and *hardware control connections*. The *system configuration* is simply a listing of hardware connections. This requires a listing (chosen from a table of instruments), from antenna to terminal device, all equipment that constitute the various sign paths. Each component connection is described by designating input and output node (and if necessary, two auxiliary nodes).

The *hardware control connections* simply designate which of the components described in the *system configuration* can be controlled and/or queried about its state. Even though a user may manually control a piece of equipment, the software will need to know about its connections to query its state (when possible) for storage of system configuration information. Included in the *hardware control connections* are the type of bus and the bus addresses - whether it be GPIB, RS232, internet, etc. (Note: the current plan is to provide the capability for all equipment to be controlled via internet connections.). Also provided within this feature, is the ability to query each pieces of equipment to determine if the hardware setup truly matches the description.

The user has the mechanism for designating the *data output path*. The user can also designate a *prefix* to be added to the file name. The default prefix is the date. This prefix is followed by a sequential file number (e.g., 010702_28, where “28" is the file number). This information is stored so that with each new software startup, the user simply needs to confirm the path and prefix, or change them if necessary.

Upon startup, the user confirms or changes various *miscellaneous survey information (electronic notebook)* such as comments, survey location, site, route, antenna heights, transmitter characteristics, etc., that are pertinent to all data contained in the data file. This is provided in a form format, where the user has the option of checking information that will be included in the data file header. With each checked category, the user is provided with a mechanism for entering the information.

The user has the option of confirming or providing various *location and vehicle bearing information*. This includes the option to manually enter the coordinate location and bearing, leaving it blank, or designating various pieces of equipment (such as GPS and dead reckoning) for automated coordinate and position determination. The user also has the option of determining (or manually entering) the location and bearing and retaining the static information throughout the measurements or dynamically determining the coordinate and bearing information using an automated means throughout the measurements. These settings will be retained and can be confirmed or changed with each software startup.

As described above the user is given the option of changing or confirming the system configuration. In doing so, the user may need to enter various discrete components such as filters, amplifiers, AGCs, log amplifiers, noise diodes, etc. Therefore, it is necessary to provide a database containing the necessary specifications for each component used. This *device characteristics database manager* provides a mechanism for storing information about each new component that can be used in the signal path. Each type of component will have its own associated form requesting specific information. For instance, each AGC and log amplifier may need an associated look-up table; filters may need bandwidth, insertion loss, and center frequency. In addition, some components may be grouped into devices such as the front-end preselector box.

2) **Performing Measurements**

The measurements module provides three mechanisms for gathering data: A) automated or semi-automated measurements executed via a scheduler using band or channel events B) automated or semi-automated measurements set up and executed in an interactive mode, and C) a fully manual measurement for the purpose of data dumping (screen dumping or streaming of digital data).

Automated and Semi-automated Measurements Using a Scheduler

Event-based measurement are used to perform fully automated or semi-automated measurements for either sequenced measurements, or scheduled measurements. For *sequenced measurements*, execution of multiple events is determined by sequential ordering; and for *scheduled measurements*, execution of multiple events is determined by sequential ordering, as well as, triggering by time, position, or incident, and weighted by priority. Sequenced measurements are a subset of scheduled measurements. An *event* is a specific measurement (methods) with specific settings (attributes). A *band event* is described in terms of the methods and attributes for an entire frequency band, and a *channel event* is described in terms of the methods and attributes for each channel within a band. The difference between band event and channel event is simply a matter of semantics and is a matter of how refined the user wishes to define an event. *Fully automated measurements* require the software to control each piece of equipment that it is capable of controlling; *semi automated measurements* allow the user to control some of the equipment settings while having the software control all other pieces of equipment.

The user defines 2 characteristics of events using an event editor. 1) measurement type and 2) measurement parameters. Once an event is defined, the individual events are stored in an *event file* which can be loaded into the scheduler at the users request. Because these files may describe measurements which can be performed using different system configuration, the event does not describe the control of all pieces of equipment along the path - only those that are essential to the measurement. Along the same lines, the event does not designate the specific model of an essential equipment type. NOTE: Measurement method modules and script files will designate the essential equipment categories required for the measurement.

The user specifies the *measurement type* from a list of possible choices. This includes any of the measurements described in the Functional Measurements Requirements document, any script defined measurements, direction finding or antenna pointing procedures, as well as, calibrations, YIG tuning procedures, calibration purge, coordinate or speed determination, temperature reading, etc. The calibration purge is provided for the user to periodically remove all calibration files that are no longer required.

Measurement Parameters: the user will be queried for the settings on each essential instrument that is designated to be controlled by the computer. Forms listing all the required input for instrument settings will be specific to the event type.

The user loads the individual events into a schedule editor and associates the event with 7 possible scheduling parameters: 1) triggering, 2) repetition 3) prioritization, 4) signal path, 5) extraneous measurements, 6) associated calibrations (by pointing to event files used for calibration), and 7) preselector settings. The combination of event and the 7 associated parameters create what is termed an “elaborated event”. The elaborated events are, in turn, saved in individual files called elaborated event files which can be referenced for future schedule editing.

Both events and blocks (but not events within a block) can have an associated trigger, repetition, and/or prioritization tag. *Trigger tags* are used to designate that an event or sequential block event are to be initiated when a specified trigger occurs. Triggers include time occurrence (a specific time or time period has passed), incident (e.g., a signal has been identified), or position occurrence (a specific coordinate region has been entered, or a distance has been traveled). *Repetition tags* are used to designate a repetition of an event or sequential block event when the event queue is empty. A *quota* (any number from 1 to infinity) determine the number of repetitions or triggered events to occur. Trigger tags and repetition tags are mutually exclusive and therefore, cannot both be applied to an event. *Prioritization tags* designate priority when two or more events or sequential block events are scheduled to occur at the same time.

Signal path: So that the software can route the signal, the user is queried regarding the proper signal path for the event in the context of the current system configuration. Signal path designation will *not* include the preselectors or the spectrum analyzer. Preselector paths will be automatically determined by the software, based on frequency ranges. Spectrum analyzer paths will be automatically determined by the measurement device. If it is determined (by the software) that different paths through the preselectors are required during a frequency sweep, the event will be broken into segments corresponding to the frequency regions associated with the various preselector paths.

Extraneous measurements: the user has the option of associating extraneous measurements such as temperature, rain rate, etc., with the specific measurement event. Results are then stored in the record header along with the data.

Each calibration performed for a specific frequency range and signal path is stored in a file. The user, in turn, links *associated calibrations* to each measurement event by assigning calibration event files or designating that no calibration is necessary. It is the users responsibility to see that all of the appropriate calibrations are performed prior to the measurement.

If one or more preselector units are in the signal path, the user is required to designate (for each preselector) whether the attenuation is to be set automatically or manually and whether the subpath within the preselector is to be set automatically or manually. If manual attenuation-setting is requested, the user is prompted for the desired value. If manual path-setting is requested, the user is prompted for the specific path.

The different elaborated events are either created in the schedule editor (and saved as a file) or referenced from a previously existing elaborated event file. These elaborated events are sequentially ordered to produce the *event schedule*. The schedule is simply an array of strings listing the elaborated event files in the order of which they are designated, and this array of strings is, in turn, stored in a file called a *schedule file*. It is possible to group various elaborated events together and associate triggering, repetition, and prioritization tags to the group as a whole. This is represented in the schedule array as a combination of the different elaborated event files, separated by commas, and placed in a single string. At the end of the string, in parenthesis, is the name of an *elaborated group file* containing the associated tags: priority, repetitions, and triggering. If the individual events of the group have their own tags, these are deactivated, and control is passed to the tags associated with the group as a whole.

The different elaborated events are executed in the following manner: all non-triggered events are loaded into an elaborated event queue in the order that they are listed by the schedule. Any triggered event or sequential block event that has met the trigger criterion, the associated calibrations have been performed, and has not met the quota is loaded into the front of the queue immediately behind any events or sequential block events of higher priority. If a sequential block event is partially executed (only some of the events are executed) and is of equal or less priority rank than the triggered event, the triggered event is loaded at the top of the queue for execution immediately after the currently executing event. When the event queue is empty, any events with a repetition tag that has not met its quota is reloaded into the event queue. Event triggering conditions are monitored by a software package executing on a separate computer and then relayed to the acquisition software via digital I/O lines. Upon meeting the conditions for triggering, the digital line is held high until the acquisition software acknowledges receipt of the trigger message.

The scheduler file name and the event name are stored in the record header of each data set associated with the scheduler; this allows re-measurement upon examination of data. The user has a choice of which scheduler file to use for implementation. As the execution of events occurs, the history of execution is stored in an external file so that, if the process is interrupted, the program may continue where it left off; the user has the option, when starting an event-driven measurement, to start anew or to continue with a measurement as determined from the history-of-execution file.

Interactive Automated and Semi-automated Measurements

Interactive automated and semi-automated measurements are the same as those described for scheduled events except that only one automated/semi-automated measurement is performed and the user is provided with an interactive interface for setting measurement parameters, and signal-path settings. The interface can also display data as it is acquired, and the user can interactively pause, stop, or continue measurements.

Fully Manual Measurements

Manual measurements with a data dump give the user the option of manually controlling equipment, either locally or remotely through virtual instrument panels, and then dumping data (screen dumping or streaming of digital data) directly to an output file. Associated with each manual measurement are five procedures: 1) signal path designation, 2) calibration, 3) YIG tuning, 4) signal direction finding or antenna pointing, and 5) designation and setup of data source.

The manual measurements are associated with a *signal path* that describes the route in which the signal travels. The user may choose from the database of discrete components, or certain instruments such as spectrum analyzer, tuner, or digitizer. The signal path is stored with the data in the output data file, as well as, stored in a signal-path file that can be accessed for future path designation.

Prior to the manual measurement, the user may perform a manual or automated *calibration* procedure through the designated signal path, and a designated frequency range Each calibration performed for a specific frequency range and signal path is stored in a file and assigned a file name that identifies the specific calibration. Prior to performing the data dump the user can associate specific calibration files with the data about to be collected. The calibration data is then stored in an output file along with the other information acquired during the data dump. Any time a new calibration is performed, a new output data file is opened and all the cal data is placed in the header.

The user has the option of performing a *YIG tuning* procedure - either automatic or manual. The YIG tuning procedure also has the option of locking the YIG to a specified center frequency and/or adding a frequency offset during the course of frequency sweeps. As described above, the automated YIG tuning procedure can also be designated as an event in the event-driven measurements.

Automated *signal direction finding or antenna pointing* procedures can likewise be performed by the user prior to manual measurements. As described above, these procedures can be entered as an event for event-driven measurements.

So that the software can determine which measurement module to utilize during the data dump, the user is required to *designate the terminal device*. Once designated, the user is provided with a virtual panel in which preselector, tuner, spectrum analyzer, and/or terminal device settings can be changed (if the appropriate control connections are established). In addition, the user is provided the opportunity to enter comments. The calibration file and signal-path file (or other extraneous measurements) can be linked to the manual measurement; and data can be acquired from the terminal device. All current calibrations files are stored in the file as records (when the file is created) and are pointed to by the individual measurement record headers. The user can continue to add data dumps to the file, each one with possible links to calibration and signal-path files.

Data Output

Any time a new calibration is performed, or the maximum number of records has occurred, or the user designates a new file through the initialization process (described above), or the *miscellaneous survey information* (described above) is changed, a new output data file is opened and all the cal data is placed in the header; otherwise, data continues to be appended to the most recent data file, irrespective of whether there is a mixture of scheduled measurements with different scheduler files, interactive automated measurements, and/or fully manual measurements with data dump; this is true even if the executable is terminated and re-started. It is, therefore, required that the software keep track of the most recent data file name and path. The coordinate and antenna bearing information is placed in the record header as designated by the user entered *logistical information* described above. All measurement parameters such as number of sweeps, step size, etc., are also stored in the measurement record header; also included are user-input comments specific to the record. Any time a component (including YIG) is designated as having *hardware control connections* (described above), the configuration parameters are stored in the record *instrument configuration* section described in the file structure document.

3) **Retrieving Data**

The data retrieval module provides the user with a mechanism to view recorded data. There are four basic components to data retrieval: 1) data inspection, 2) conversion to ASCII output, 3) re-measurement, and 4) appending comments.

The user can *inspect data* by two different means: 1) by examining the raw data in ASCII format, or 2) by plotting the data. The ASCII format output simply shows the raw data values placed in properly labeled columns; included is text showing header information such as signal path and survey description. Plotting of data is accomplished by associating a plotting routine that is appropriate to the measurement type stored in a specific record within the data file. Not all data will have an associated plotting routine. Upon examination of the data, the user has the option of appending or changing contents in the comments field of the record header.

The data can be *converted to an ASCII output file* if the user wishes. The format is the same as the ASCII text described above for viewing the data.

Re-measurement simply means that the user has the option of using the stored information in the data file to set up the measurement equipment with the same settings for a repeat measurement. For anything other than fully-automated measurements, the setup will only be partial, and in those cases, it is necessary for the user to complete the setup. For event-driven measurements, the name of the associated *scheduler file* is stored in the data file header; when the user requests the re-measurement of an event-driven measurement, the software simply re-loads the scheduler file and gives the user the option of executing “as is” or editing prior to execution.

The user is given the option of *appending comments* in a batch mode, whereby the user can write a comment and have it appended to specific groups of records within a file. The user will also have the option of appending or changing comments which apply to the file in general by editing the comments located in the file header.

4) **Editing and Testing Script Language Routines**

Script language provides the user with a means to build custom routines for specialized signal measurements or to build prototypal routines used for proof-of-concept during development. This module utilizes a standard editor, whereby the user enters specifically defined command text that can be interpreted for execution on the targeted instruments. The language is generic for the different instrument types (e.g. spectrum analyzer, digital oscilloscope, etc.,) and is interpreted in relation to the specific instrument model targeted for execution. (However, there may be language exceptions for specialized features on the targeted models.) Text is saved in an ASCII file and can then be called for execution.

Prior to execution within this module, the user designates which instruments (that have an associated control bus) are to be controlled by the computer during the test. Once the script file is tested, it can be used in the event editor to designate a specific measurement event. The name of the script file used for measurement will be stored in the data output file for future reference.